CLAIMS

1. In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching in parallel for a plurality of prime number values substantially simultaneously, comprising the steps of:

randomly generating a plurality of k random odd numbers each providing a prime number candidate; and

performing at least one primality test on each of said candidates, each of said primality tests including an associated exponentiation operation executed by an associated one of the exponentiation units, said exponentiation operations being performed by said associated exponentiation units substantially simultaneously.

2. In a prime number generating system as recited in claim 1 wherein said plurality of k randomly generated numbers are expressed as $n_{0,0}$, $n_{1,0}$, … $n_{((k-1)),0}$, further comprising the steps of:

determining a plurality of y additional odd numbers based on each one of the randomly generated numbers $n_{0,0}$, $n_{1,0}$, … $n_{(k-1),0}$ to provide $\mathbf{(k \times y)}$ additional prime number candidates $(n_{0,1}, n_{0,2}, \ldots n_{0,y})$, $(n_{1,1}, n_{1,2}, \ldots n_{1,y})$, …$(n_{(k-1),1}, n_{(k-1),2}, \ldots n_{(k-1),y})$ thereby yielding a total number of $\mathbf{(k \times (y+1))}$ prime number candidates;

wherein said step of performing includes performing a primality test on each of said total number $\mathbf{(k \times (y+1))}$ of candidates, each of the plurality of $\mathbf{(k \times (y+1))}$ primality tests including an associated exponentiation operation executed by an associated one of a plurality of $\mathbf{(k \times (y+1))}$ of the exponentiation units, said exponentiation operations being performed by said plurality of $\mathbf{(k \times (y+1))}$ exponentiation units substantially simultaneously.

3. In a prime number generating system as recited in claim 2 wherein each of said plurality of prime number values being searched for has a specified length, and wherein said plurality of y additional odd numbers defines an interval that is selected relative to said specified length.

4. In a prime number generating system as recited in claim 2 wherein said step of determining a plurality of y additional odd numbers based on each one of the randomly generated numbers $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$ includes successively adding two to each of said randomly generated odd numbers $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates expressed as $(n_{0,1} = n_{0,0}+ 2 , n_{0,2} = n_{0,0}+ 4, \ldots n_{0,y} = n_{0,0} + (y\cdot2)), (n_{1,1} = n_{1,0}+ 2 , n_{1,2} = n_{1,0}+ 4, \ldots n_{1,y} = n_{1,0} + (y\cdot2)), \ldots (n_{(k-1),1} = n_{(k-1),0}+ 2 , n_{(k-1),2} = n_{(k-1),0}+ 4, \ldots n_{(k-1),y} = n_{(k-1),0} + (y\cdot2))$.

5. In a prime number generating system as recited in claim 1 wherein said step of performing includes performing a plurality of t primality tests on each one of said plurality of k randomly generated numbers, each of the plurality of (k x t) primality tests including an associated exponentiation operation executed by an associated one of a plurality of (k x t) of the exponentiation units, said exponentiation operations being performed by said associated exponentiation units substantially simultaneously.

6. In a prime number generating system as recited in claim 2 wherein said step of performing includes performing a plurality of **t** primality tests on each of said **(k x (y+1))** prime number candidates, each of the plurality of **(k x (y+1) x t)** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(k x (y+1) x t)** of the exponentiation units, said exponentiation operations being performed by said plurality of **(k x (y+1) x t)** exponentiation units substantially simultaneously.

7. In a prime number generating system as recited in claim 1 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing at least one primality test on each of said sieved number **s** of candidates, each of the plurality of **s** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously.

8. In a prime number generating system as recited in claim 7 further comprising the steps of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate minus one and said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing at least one primality test.

9. In a prime number generating system as recited in claim 2 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing at least one primality test on each of said sieved number **s** of candidates, each of the plurality of **s** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously.

10. In a prime number generating system as recited in claim 1 further comprising the step of:

sieving said prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality test on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

11. In a prime number generating system as recited in claim 2 further comprising the step of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

12. In a prime number generating system as recited in claim 1 wherein said step of performing at least one primality test includes performing a Fermat type primality test.

13. In a prime number generating system as recited in claim 1 wherein said step of performing at least one primality test includes performing a Miller-Rabin type primality test.

14. In a prime number generating system as recited in claim 1 wherein said step of randomly generating a plurality of k random odd numbers further includes:

defining a length L for each of the plurality of k random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

15. In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching in parallel for a plurality of prime number values simultaneously, comprising the steps of:

randomly generating at least one random odd number providing a prime number candidate;

determining a plurality of **y** additional odd numbers based on said at least one randomly generated odd number to provide **y** additional prime number candidates, thereby providing a total number of **y+1** candidates;

performing at least one primality test on each of said **y+1** candidates, each of the **y+1** primality tests including an associated exponentiation operation executed by an associated one of **y+1** of the exponentiation units, said **y+1** exponentiation operations being performed by said associated **y+1** exponentiation units substantially simultaneously.

16. In a prime number generating system as recited in claim 15 wherein said at least one randomly generated odd number is expressed as $n_{0,0}$, and wherein said step of determining a plurality of y additional odd numbers based on said randomly generated odd number $n_{0,0}$ includes successively adding two to said randomly generated odd number $n_{0,0}$ to provide **(y+1)** additional prime number candidates expressed as $(n_{0,1} = n_{0,0}+ 2 , n_{0,2} = n_{0,0}+ 4, \ldots n_{0,y} = n_{0,0} + (y\cdot 2))$.

17. In a prime number generating system as recited in claim 15 wherein said step of performing includes performing a plurality of **t** primality tests on each of said **(y+1)** prime number candidates, each of the plurality of **((y+1)** x **t)** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **((y+1)** x **t)** of the

exponentiation units, said exponentiation operations being performed by said plurality of **((y+1)** x **t)** exponentiation units substantially simultaneously.

18. In a prime number generating system as recited in claim 15 further comprising the step of sieving said **y+1** candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates.

19. In a prime number generating system as recited in claim 18 further comprising the step of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate minus one and said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing at least one primality test.

20. In a prime number generating system as recited in claim 15 further comprising the step of:

sieving said **y+1** candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality test on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, the **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

21. In a prime number generating system as recited in claim 15 wherein said step of performing at least one primality test includes performing a Fermat type primality test.

22. In a prime number generating system as recited in claim 15 wherein said step of performing at least one primality test includes performing a Miller-Rabin type primality test.

23. In a prime number generating system as recited in claim 15 wherein said step of randomly generating at least one random odd number further includes:

defining a length L for each of the plurality of k random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

24. In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching in parallel for a plurality of prime number values simultaneously, comprising the steps of:

randomly generating at least one random odd number providing a prime number candidate; and

testing the primality of said candidate by performing a plurality of **t** primality tests on said candidate, each of the plurality of the **t** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **t** of the exponentiation units, said exponentiation operations being performed by said plurality of **t** exponentiation units substantially simultaneously.

25. In a prime number generating system as recited in claim 24 further including the step of sieving said candidates by performing a small divisor test on each of said candidates in order to

eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates.

26. In a prime number generating system as recited in claim 25 further including the step of::

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate minus one and said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing at least one primality test.

27. In a prime number generating system as recited in claim 24 wherein said step of testing the primality of said candidate further includes:

sieving said candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

performing an associated first one of said **t** primality test on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional ones of said t primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

28. In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of

searching in parallel for a plurality of prime number values simultaneously, comprising the steps of:

randomly generating a plurality of k random odd numbers expressed as $n_{0,0}$, $n_{1,0}$, ... $n_{((k-1)),0}$, each said number providing a prime number candidate;

determining a plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{0,0}$, $n_{1,0}$, ... $n_{(k-1),0}$ to provide (k x y) additional prime number candidates $(n_{0,1}, n_{0,2}, \ldots n_{0,y})$, $(n_{1,1}, n_{1,2}, \ldots n_{1,y})$, ...$(n_{(k-1),1}, n_{(k-1),2}, \ldots n_{(k-1),y})$ thereby yielding a total number of (k x (y+1)) prime number candidates;

sieving said (k x (y+1)) prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number s of candidates; and

performing at least one primality test on each of said sieved number s of candidates, each of the plurality of s primality tests including an associated exponentiation operation executed by an associated one of a plurality of s of the exponentiation units, said exponentiation operations being performed by said plurality of s exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said primality test thereby yielding a remaining number r of candidates.

29. In a prime number generating system as recited in claim 28 wherein said step of determining a plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{0,0}$, $n_{1,0}$, ... $n_{(k-1),0}$ includes successively adding two to each of said randomly generated odd numbers $n_{0,0}$, $n_{1,0}$, ... $n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates expressed as $(n_{0,1} = n_{0,0} + 2, n_{0,2} = n_{0,0} + 4, \ldots n_{0,y} = n_{0,0} + (y \cdot 2))$, $(n_{1,1} = n_{1,0} + 2, n_{1,2} = n_{1,0} + 4, \ldots n_{1,y} = n_{1,0} + (y \cdot 2))$, ... $(n_{(k-1),1} = n_{(k-1),0} + 2, n_{(k-1),2} = n_{(k-1),0} + 4, \ldots n_{(k-1),y} = n_{(k-1),0} + (y \cdot 2))$.

30. In a prime number generating system as recited in claim 28 further comprising the steps of:

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

31. In a prime number generating system as recited in claim 28 wherein said step of performing at least one primality test includes performing a Fermat type primality test.

32. In a prime number generating system as recited in claim 28 wherein said step of performing at least one primality test includes performing a Miller-Rabin type primality test.

33. In a prime number generating system as recited in claim 28 wherein said step of randomly generating a plurality of k random odd numbers further includes:

defining a length L for each of the plurality of k random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

34. In a prime number generating system as recited in claim 28 wherein k is greater than or equal to 2.

35. In a prime number generating system as recited in claim 28 further comprising the steps of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate minus one and said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing at least one primality test.

36. A prime number generating system for searching in parallel for a plurality of prime number values simultaneously, comprising:

processing means operative to randomly generate a plurality of k random odd numbers each providing a prime number candidate, and to provide at least one set of test parameters associated with a primality test to be performed on each one of said plurality of k randomly generated numbers, each said set of said test parameters including said associated randomly generated number and an associated base value; and

a plurality of exponentiation units each being communicatively coupled with said processing means, and being responsive to an associated one of said sets of test parameters, and operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said exponentiation units being operative to perform said exponentiation operations substantially simultaneously;

said processing means being responsive to said primality test result signals, and operative to process said test result signals for the purpose of eliminating randomly generated numbers declared to be composite in accordance with a search for prime number values.

37. A prime number generating system as recited in claim 36 wherein:

said processing means is operative to provide a plurality of **t** sets of test parameters associated with a plurality of **t** primality tests to be performed on each one of said plurality of k randomly generated numbers, each of the (k x t) sets of said test parameters including an associated one of said plurality of k randomly generated numbers and an associated one of a plurality of **t** base values;

said plurality of exponentiation units includes a plurality of at least **(k x t)** exponentiation units each being responsive to an associated one of said **(k x t)** sets of said test parameters, and being operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **(k x t)** exponentiation units being operative to perform said plurality of **(k x t)** exponentiation operations substantially simultaneously.

38. A prime number generating system as recited in claim 36 wherein said processing means is further operative to sieve said prime number candidates by performing a small divisor test on each of said prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates.

39. A prime number generating system as recited in claim 36 wherein said plurality of k randomly generated odd numbers are expressed as $n_{0,0}$, $n_{1,0}$, … $n_{((k-1)),0}$, and wherein:

said processing means is further operative to develop a plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{0,0}$, $n_{1,0}$, … $n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates ($n_{0,1}$, $n_{0,2}$, … $n_{0,y}$), ($n_{1,1}$, $n_{1,2}$, … $n_{1,y}$), …($n_{(k-1),1}$, $n_{(k-1),2}$, … $n_{(k-1),y}$) thereby yielding a total number of **(k x (y+1))** prime number candidates;

said plurality of exponentiation units includes a plurality of at least **(k x (y+1))** exponentiation units each being responsive to an associated one of said **(k x (y+1))** sets of said test parameters, and being operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **(k x (y+1))** exponentiation units being operative to perform the plurality of **(k x (y+1))** exponentiation operations substantially simultaneously.

40. A prime number generating system as recited in claim 39 wherein said processing means is operative to develop said plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{0,0}$, $n_{1,0}$, … $n_{(k-1),0}$ by successively adding two to each of said randomly generated odd numbers $n_{0,0}$, $n_{1,0}$, … $n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates expressed as ($n_{0,1} = n_{0,0}+ 2$, $n_{0,2} = n_{0,0}+ 4$, … $n_{0,y} = n_{0,0} + (y \cdot 2)$), ($n_{1,1} = n_{1,0}+ 2$, $n_{1,2} = n_{1,0}+ 4$, … $n_{1,y} = n_{1,0} + (y \cdot 2)$), … ($n_{(k-1),1} = n_{(k-1),0}+ 2$, $n_{(k-1),2} = n_{(k-1),0}+ 4$, … $n_{(k-1),y} = n_{(k-1),0} + (y \cdot 2)$).

41. A prime number generating system as recited in claim 39 wherein:

said processing means is operative to provide a plurality of t sets of test parameters associated with a plurality of t primality tests to be performed on each one of said plurality of $\mathbf{(k \times (y+1))}$ randomly generated numbers, each of the $\mathbf{(k \times (y+1) \times t)}$ sets of said test parameters including said associated one of said plurality of $\mathbf{(k \times (y+1))}$ prime number candidates and an associated one of a plurality of **t** base values;

said plurality of exponentiation units includes a plurality of at least $\mathbf{(k \times (y+1) \times t)}$ exponentiation units each being responsive to an associated one of said $\mathbf{(k \times (y+1) \times t)}$ sets of said test parameters, and being operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least $\mathbf{(k \times (y+1) \times t)}$ exponentiation units being operative to perform said plurality of $\mathbf{(k \times (y+1) \times t)}$ exponentiation operations substantially simultaneously.

42. A prime number generating system as recited in claim 36 wherein each of said primality tests is a Fermat type primality test.

43. A prime number generating system as recited in claim 36 wherein each of said primality tests is a Miller-Rabin type primality test.

44. A prime number generating system as recited in claim 36 wherein said processing means is operative to randomly generate said plurality of k random odd numbers by performing the steps of:

defining a length L for each of the plurality of k random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

45. A prime number generating system for searching in parallel for a plurality of prime number values simultaneously, comprising:

processing means operative to randomly generate at least one random odd number providing a prime number candidate, and to determine a plurality of y additional odd numbers based on each of said at least one randomly generated odd number to provide **y** additional prime number candidates, thereby providing a total number of **y+1** candidates, said processing means also being operative to provide at least one set of test parameters associated with a primality test to be performed on each one of said prime number candidates, each said set of test parameters including said associated prime number candidate and an associated base value; and

a plurality of exponentiation units each being communicatively coupled with said processing means, and being responsive to an associated one of said sets of test parameters, and operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said exponentiation units being operative to perform said exponentiation operations substantially simultaneously;

said processing means being responsive to said primality test result signals, and operative to process said test result signals for the purpose of eliminating randomly generated numbers declared to be composite in accordance with a search for prime number values.

46. A prime number generating system as recited in claim 45 wherein said at least one randomly generated odd number is expressed as $n_{0,0,}$ and wherein said processing means operative to determine said plurality of y additional odd numbers based on said randomly generated odd number by successively adding two to said randomly generated odd number $n_{0,0}$ to provide **(y+1)** additional prime number candidates expressed as ($n_{0,1} = n_{0,0}+ 2$, $n_{0,2} = n_{0,0}+ 4$, … $n_{0,y} = n_{0,0} + (y \cdot 2)$).

47. A prime number generating system as recited in claim 45 wherein:

said processing means is operative to provide a plurality of **t** sets of test parameters associated with a plurality of **t** primality tests to be performed on each one of said plurality of at least **y+1** randomly generated numbers, each of the **((y+1) x t)** sets of said test parameters including said associated one of said plurality of candidates and an associated one of a plurality of **t** base values;

said plurality of exponentiation units includes a plurality of at least **((y+1) x t)** exponentiation units each being responsive to an associated one of said **((y+1) x t)** sets of said test parameters, and being operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **((y+1) x t)** exponentiation units being operative to perform said plurality of **((y+1) x t)** exponentiation operations substantially simultaneously.

48. A prime number generating system as recited in claim 45 wherein said processing means is further operative to sieve said prime number candidates by performing a small divisor test on each of said prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates.

49. A prime number generating system as recited in claim 45 wherein:

said processing means is operative to generate a plurality of **k** random odd numbers each providing a prime number candidate, and to determine a plurality of y additional odd numbers based on each of said **k** random odd numbers to provide **k x y** additional prime number candidates, thereby providing a total number of at least **k x (y+1)** candidates, said processing means also being operative to provide at least one set of test parameters associated with a primality test to be performed on each one of said **k x (y+1)** prime number candidates, each said set of said test parameters including said associated prime number candidate and an associated base value; and

said plurality of exponentiation units includes a plurality of at least **k x (y+1))** exponentiation units each being responsive to an associated one of said **k x (y+1)** sets of said test parameters, and being operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **k** x **(y+1)** exponentiation units being operative to perform said plurality of **k** x **(y+1)** exponentiation operations substantially simultaneously.

50. A prime number generating system as recited in claim 45 wherein each of said primality tests is a Fermat type primality test.

51. A prime number generating system as recited in claim 45 wherein each of said primality tests is a Miller-Rabin type primality test.

52. A prime number generating system as recited in claim 45 wherein said processing means is operative to randomly generate said plurality of k random odd numbers by performing the steps of:

defining a length L for each of the plurality of k random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

53. A prime number generating system for searching in parallel for a plurality of prime number values simultaneously, comprising:

processing means operative to randomly generate a plurality of random odd numbers each providing a prime number candidate, and to provide a set of test parameters associated with a primality test to be performed on each one of said plurality of randomly generated numbers, each said set of said test parameters including said associated randomly generated number; and

a plurality of exponentiation units each being communicatively coupled with said processing means, and being responsive to an associated one of said sets of test parameters, and operative to perform an exponentiation operation based on said associated set of test parameters and an associated base value, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said exponentiation units being operative to perform said exponentiation operations substantially simultaneously;

said processing means being responsive to said primality test result signals, and operative to process said test result signals for the purpose of eliminating randomly generated numbers declared to be composite in accordance with a search for prime number values.

54. A computer readable storage medium having stored thereon encoding instructions for executing a process of searching in parallel for a plurality of prime number values simultaneously in a prime number generation system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, the process comprising the steps of:

randomly generating a plurality of k random odd numbers each providing a prime number candidate; and

performing at least one primality test on each of said candidates, each of said primality tests including an associated exponentiation operation executed by an associated one of the exponentiation units, said exponentiation operations being performed by said associated exponentiation units substantially simultaneously.

55. A computer readable storage medium as recited in claim 54 wherein said plurality of k randomly generated numbers are expressed as $n_{0,0}$, $n_{1,0}$, … $n_{((k-1)),0}$, further comprising the steps of:

determining a plurality of y additional odd numbers based on each one of the randomly generated numbers $n_{0,0}$, $n_{1,0}$, … $n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates ($n_{0,1}$, $n_{0,2}$, … $n_{0,y}$), ($n_{1,1}$, $n_{1,2}$, … $n_{1,y}$), …($n_{(k-1),1}$, $n_{(k-1),2}$, … $n_{(k-1),y}$) thereby yielding a total number of **(k x (y+1))** prime number candidates;

wherein said step of performing includes performing a primality test on each of said total number **(k x (y+1))** of candidates, each of the plurality of **(k x (y+1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(k x (y+1))** of the exponentiation units, said exponentiation operations being performed by said plurality of **(k x (y+1))** exponentiation units substantially simultaneously.

56. A computer readable storage medium as recited in claim 55 wherein said step of determining a plurality of y additional odd numbers based on each one of the randomly generated numbers $n_{0,0}$, $n_{1,0}$, … $n_{(k-1),0}$ includes successively adding two to each of said randomly generated odd numbers $n_{0,0}$, $n_{1,0}$, … $n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates expressed as $(n_{0,1} = n_{0,0}+2, n_{0,2} = n_{0,0}+4, \ldots n_{0,y} = n_{0,0} + (y \cdot 2)), (n_{1,1} = n_{1,0}+2, n_{1,2} = n_{1,0}+4, \ldots n_{1,y} = n_{1,0} + (y \cdot 2)), \ldots (n_{(k-1),1} = n_{(k-1),0}+2, n_{(k-1),2} = n_{(k-1),0}+4, \ldots n_{(k-1),y} = n_{(k-1),0} + (y \cdot 2))$.

57. A computer readable storage medium as recited in claim 54 wherein said step of performing includes performing a plurality of t primality tests on each one of said plurality of k randomly generated numbers, each of the plurality of (k x t) primality tests including an associated exponentiation operation executed by an associated one of a plurality of (k x t) of the exponentiation units, said exponentiation operations being performed by said associated exponentiation units substantially simultaneously.

58. A computer readable storage medium as recited in claim 55 wherein said step of performing includes performing a plurality of **t** primality tests on each of said **(k x (y+1))** prime number candidates, each of the plurality of **(k x (y+1) x t)** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(k x (y+1) x t)** of the exponentiation units, said exponentiation operations being performed by said plurality of **(k x (y+1) x t)** exponentiation units substantially simultaneously.

59. A computer readable storage medium as recited in claim 54 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing at least one primality test on each of said sieved number **s** of candidates, each of the plurality of **s** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously.

60. A computer readable storage medium as recited in claim 55 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing at least one primality test on each of said sieved number **s** of candidates, each of the plurality of **s** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously.

61. A computer readable storage medium as recited in claim 54 further comprising the step of:

sieving said prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality test on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

62. A computer readable storage medium as recited in claim 55 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units substantially simultaneously in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** of the exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units substantially simultaneously in order to eliminate further candidates revealed to be composite numbers.

63. A computer readable storage medium as recited in claim 59 further comprising the steps of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of each said prime number candidate minus one and said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing at least one primality test.